

Application Layer Congestion Control for Network-Aware Telehaptic Communication

Vineet Gokhale, Indian Institute of Technology Bombay
 Jayakrishnan Nair, Indian Institute of Technology Bombay
 Subhasis Chaudhuri, Indian Institute of Technology Bombay

In a real-world network, shared by several users, telehaptic applications involving delay-sensitive multimedia communication between remote locations demand distinct Quality of Service (QoS) guarantees for different media components. These QoS constraints pose a variety of challenges, especially when the network cross-traffic is unknown and time-varying. In this work, we propose an application layer congestion control protocol for telehaptic applications operating over shared networks, termed as *dynamic packetization module* (DPM). DPM is a lossless, network-aware protocol which tunes the telehaptic packetization rate based on the level of congestion in the network. To monitor the network congestion, we devise a novel *network feedback module*, which communicates the end-to-end delays encountered by the telehaptic packets to the respective transmitters, with negligible overhead. Via extensive simulations, we show that DPM meets the QoS requirements of telehaptic applications over a wide range of network cross-traffic conditions. We also report the qualitative results of a real-time telepottery experiment with several human subjects, which reveal that DPM preserves the quality of telehaptic activity even under heavily congested network scenarios. Finally, we compare the performance of DPM with existing telehaptic communication protocols, and conclude that DPM outperforms other techniques.

Additional Key Words and Phrases: Telehaptic communication, congestion control, application layer, dynamic rate adaptation, QoS, multimedia

1. Introduction

Telehaptic applications, such as telesurgery [Anderson and Spong 1989], involve long distance transfer of haptic-audio-visual information between distantly located users. The performance of a telehaptic activity is governed by a set of Quality of Service (QoS) parameters, specific to each type of media. According to [Matsumotoy et al. 2000; Miras et al. 2002; Marshall et al. 2008], the QoS one-way delay and jitter specifications for multimedia are, respectively, as follows: video - 400 ms and 30 ms, audio - 150 ms and 30 ms, haptic - 30 ms and 10 ms.¹ Non-conformance to the above conditions leads to a degraded human perception, and can potentially introduce ill-effects on the quality of the telehaptic activity [Jay et al. 2007]. In particular, the haptic QoS violation results in destabilizing the haptic global control loop [Ferrell 1965; Anderson and Spong 1989], and a deteriorated perception of haptic objects. Hence, multimedia data reception and display within the prescribed QoS deadline plays a pivotal role in determining the stability, and the overall performance of a telehaptic task.

In a shared network, like the Internet, a telehaptic source shares the network resources with other concurrent traffic streams. As a result, the intensity of the cross-traffic encountered by a telehaptic stream on a shared network is both unknown as well as time-varying. In such a scenario, the transmission of telehaptic data in a *network-oblivious* manner can be highly sub-optimal. In particular, at times when the

¹A few works reveal that in case of audio and video, the individual human sensitivities can be time-varying [Ntonia et al. 2014; Freeman et al. 2013]. However, we note that for the purpose of network protocol design, one has to work with conservative and universal QoS specifications.

network is severely congested, a network-oblivious telehaptic stream may suffer a high delay and packet loss, leading to QoS violations. Note that this is all the more likely in resource constrained networks, such as wireless ad-hoc networks. On the other hand, at times when the network is lightly loaded, it may be feasible to transmit telehaptic data at its peak rate. The above discussion motivates the need for a *network-aware* telehaptic transmission scheme. In this paper, we propose such a scheme, which monitors network conditions in real-time and adapts the telehaptic data rate accordingly to achieve congestion control.

1.1. Contributions of the Article

In this article, we focus on a point-to-point telehaptic communication involving two telehaptic nodes connected through a shared network. Specifically, we propose a delay-based network adaptive communication protocol for telehaptic applications. Our contributions are the following.

- (1) For monitoring the network state, we introduce a *network feedback module*, in which the telehaptic receivers inform the corresponding transmitters of the end-to-end delays encountered by the packets. The end-to-end delays as measured by a receiving node are piggybacked on telehaptic data packets on the reverse channel, enabling real-time feedback of network delays to the transmitting node by adding a negligible amount of overhead on the network (3 bytes per packet).
- (2) We design the *dynamic packetization module* (DPM) for telehaptic data rate adaptation, based on the network state feedback from the network feedback module. DPM is motivated by the following observation: Under telehaptic data transmission at the default packetization rate of 1000 packets/sec, the overhead due to packet headers from various layers accounts for almost half the total telehaptic traffic. DPM is a lossless transmission scheme that opportunistically varies the telehaptic packetization rate based on the current network state. Specifically, in congested networks, DPM dynamically merges successive telehaptic samples into a single packet, thus adapting the overall transmission rate depending on the available network capacity.
- (3) We evaluate the proposed telehaptic transmission scheme via extensive simulations as well as human subjective tests through a real-time telepottery experiment. Our simulations reveal that DPM meets the telehaptic QoS specifications even under extremely congested network settings. Our subjective tests confirm that DPM provides a seamless telehaptic user experience in a congested network. Finally, we compare DPM with other recently proposed telehaptic communication protocols and demonstrate that DPM outperforms these protocols with respect to QoS compliance.

1.2. Related Work

There have been several attempts to address the problem of large telehaptic bandwidth requirement. The standard input and output update rate of the haptic signal is 1 kHz. In order to reduce the packetization delay encountered by the haptic samples, the conventional approach follows fixed haptic packetization at 1 kHz (1 packet per sample) for transmission over the network. This approach is highly bandwidth demanding, and is not friendly to other network users. To counter this issue, the works in [Hinterseer et al. 2005; Clarke et al. 2006; Hinterseer and Steinbach 2006; Hinterseer et al. 2008; Dabeer and Chaudhuri 2011; Sakr et al. 2011; Bhardwaj et al. 2013] explored *adaptive sampling* which exploits the perceptual limitation of the human haptic system, of not perceiving certain haptic samples, to achieve lossy haptic signal compression. A *just noticeable difference* (JND)

metric adaptively marks the haptic samples that are not perceivable by the human users. The communication system refrains from transmitting such samples, thereby reducing the telehaptic data rate. The missing haptic samples are then reconstructed at the receiver using standard extrapolation techniques listed in [Gokhale et al. 2013]. However, critical operations, like telesurgery, necessitate accurate replication of the surgeon's hand movements. In such scenarios, a minor loss of precision due to adaptive sampling could result in potentially irreparable damage. Also, a teleoperator, such as a robotic device, could practically sense all haptic samples; in such cases, adaptive sampling discards perceptually significant samples. Another networking-related issue with adaptive sampling is the following. The *instantaneous* source rate of adaptive sampling depends purely on the speed of haptic interaction, and can at times far exceed the *average* source rate. As a result, provisioning the network for the average source rate can lead to serious QoS violations; this is demonstrated in Section 4.3. In other words, adaptive sampling does not provide any real economies with respect to network bandwidth requirement – one needs to provision network capacity for the *peak* telehaptic data rate in order to avoid QoS violations.

Several application layer protocols have been specifically designed for telehaptic communication. ALPHAN: Application Layer Protocol for HAptic Networking, proposed in [Al Osman et al. 2007], implements haptic and graphic data communication at the packetization rate of 1 kHz. AdMux: Adaptive Multiplexer [Eid et al. 2011] proposes a statistical multiplexing scheme for scheduling haptic-audio-video packet transmission based on the QoS requirements and changing network behavior. Haptics over Internet Protocol (HoIP) for point-to-point communication, proposed in [Gokhale et al. 2015], addresses media multiplexing and telehaptic communication involving haptics, audio and video data. The above mentioned protocols carry out telehaptic transmission at the peak rate, and hence do not address the problem of congestion control. In [Cizmeci et al. 2014], the authors consider visual-haptic multiplexing over constant-bitrate (CBR) communication links, employing adaptive sampling for haptic signal compression. However, the drawbacks of adaptive sampling mentioned previously apply here; see Section 4.3 for a demonstration.

The work in [Fujimoto and Ishibashi 2005], which explores the possibility of merging multiple haptic samples in a packet to reduce the telehaptic data rate. In contrast with the scheme proposed in this paper, the scheme in [Fujimoto and Ishibashi 2005] always combines a fixed number of haptic samples, irrespective of the network conditions. Note that this implies unnecessary packetization delay even when the network is uncongested. Moreover, the authors showed in a particular setting that a packetization interval of 8 ms results in a satisfactory user performance. On the contrary, we demonstrate (see Figure 4) that the packetization intervals greater than 4 ms result primarily in increasing end-to-end delays, without any substantial reduction in the telehaptic data rate.

Note that the above mentioned proposals are all *network-oblivious*, i.e., they do not adapt the telehaptic transmission rate based on network conditions. The literature provides a few works that have considered *network-aware* telehaptic rate adaptation. We discuss these next.

In [Lee and Kim 2007], the authors propose a network adaptation scheme for merging haptic samples based on packet losses arising out of congestion. Such a scheme is reactive to network congestion, in the sense that data rate reduction is activated only after detecting persistent packet losses. Clearly, such a loss-based congestion control mechanism is not suitable for highly delay-sensitive telehaptic applications. We note that [Lee and Kim 2007] does not provide much detail about the rate adaptation mechanism itself; also, the effects of this rate adaptation on other concurrent network flows are not analyzed.

The authors in [Wirz et al. 2008] propose the first delay-sensitive haptic communication protocol named Efficient Transport Protocol (ETP). ETP detects congestion based on round-trip-time (RTT) measurements. Once congestion is detected, ETP reduces the telehaptic data rate by increasing the interpacket gap, i.e., by *downsampling* the haptic signal. In contrast, the protocol proposed in this paper preserves the fidelity of the haptic signal, adapting instead the packetization rate based on the congestion level in the network.

The paper most closely related to ours is [Kokkonis et al. 2015], which proposes NAFCAH: Network Adaptive Flow Control Algorithm for Haptic data. Like DPM, NAFCAH adapts the number of haptic samples to be merged into a packet on the forward channel based on network conditions. However, there are two crucial differences between NAFCAH and DPM. First, when congestion is detected, NAFCAH decreases its transmission rate in stages. In contrast, DPM responds to congestion with an aggressive rate reduction, which enables network buffers to get flushed quickly, minimizing the possibility of QoS violations. Second, NAFCAH monitors congestion based on RTT measurements. However, under asymmetric network conditions, RTT may not provide an accurate estimate of the (one-way) delay on the forward channel. In contrast, DPM estimates the delay on the forward and backward channels separately. The performance implications of the above differences are demonstrated in Section 4.3.

The authors of [Gokhale et al. 2016] propose a network-aware opportunistic adaptive haptic sampling mechanism, wherein the Weber threshold is adapted based on the congestion level in the network. Note that the limitations of adaptive sampling discussed earlier apply to this work as well.

Finally, we contrast our work with the Real-time transport protocol (RTP) [Jacobson et al. 2003], which is the most commonly used protocol for audio/video streaming and has also been recommended by some researchers for telehaptic communication; see for example [Steinbach et al. 2012]. RTP uses report-based notification for monitoring the network conditions at regular intervals of time. The multimedia receiving agent sends RTP Control Protocol (RTCP) receiver reports to the transmitters, for QoS monitoring, once in every 500 ms [Tos and Ayav 2011]. However, as we demonstrate in Section 4.3, such sparse feedback is insufficient for telehaptic applications, which are sensitive to network changes that occur over a timescale of tens of milliseconds.

1.3. Organization of the Article

The article is organized as follows. In Section 2, we describe the configuration of a typical telehaptic environment, and explain in detail the design and working of the proposed telehaptic communication framework. In Section 3, we discuss the setup for simulations and the real-time telepottery experiment. Section 4 presents the findings of the experiments, and in Section 5, we state our conclusions.

2. Design of Telehaptic Communication Framework

In this section, we explain the standard telehaptic setting on a shared network, and describe the techniques proposed in this article for a lossless, network-aware, adaptive telehaptic data communication.

2.1. Typical Telehaptic Environment

We consider a typical point-to-point telehaptic application, like telesurgery, running on a shared network as shown in Figure 1. The operator (OP) acts as the master and sends the current position and velocity commands to the teleoperator (TOP). In response, the TOP, acting as the slave, transmits force information to the OP, in addition to auditory and visual data. The channels on which the operator and teleoperator

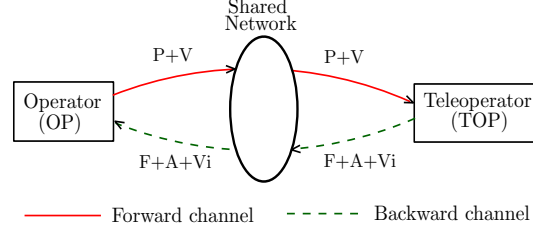


Fig. 1: A diagrammatic representation of master-slave based telehaptic setting showing the telehaptic data flow in a shared network. Data flow notations: P - position, V - velocity, F - force, A - audio, Vi - video.

transmit telehaptic data are called forward and backward channels, respectively. Note that the telehaptic traffic is bidirectional and asymmetric in nature. Moreover, the forward and backward channel are also asymmetric; in general, they may differ with respect to routing paths, capacity, as well as cross-traffic. Our network feedback module, described in Section 2.3, estimates congestion on the forward and the backward channels separately. Finally, we remark that the particular master-slave setup depicted in Figure 1 is assumed only for concreteness in exposition. Our proposed telehaptic communication protocol works in any general telehaptic application.

2.2. Media Multiplexing Framework

In this section, we describe our media multiplexing framework. Multiplexing the media frames appropriately from the different capturing devices and forwarding them to the transmitter is a critical task in any network based real-time interactive application, since it directly influences the QoS adherence of the respective media. The authors in [Cizmeci et al. 2014] rightly explain the importance of splitting a large video frame into smaller parts for transmission. Naturally, if a large video frame is transmitted in a single packet, it would clog the network for a long time, thereby delaying the subsequent haptic/audio samples substantially. The media multiplexing framework proposed here is an adaptation of that in [Cizmeci et al. 2014; Gokhale et al. 2015].

Our media multiplexer works in synchronization with the sampling of the haptic signal, which we assume occurs at the default rate of 1 kHz. Each time a haptic sample is generated, our multiplexer generates a telehaptic fragment of size p , which contains the latest haptic sample, as well as audio/video data as explained below.²

The telehaptic payload generation rate, denoted by D (in kbps), is expressed in terms of the individual media parameters as

$$D = (f_h \cdot s_h + f_a \cdot s_a + f_v \cdot s_v) \cdot (8/1000), \quad (1)$$

where f_h , f_a and f_v represent the sampling rates of haptic, audio and video data, respectively in samples per second.³ s_h , s_a and s_v represent the size of a single haptic sample, audio and video frame, respectively in bytes. In order to maintain equilibrium between the payload generation and the multiplexing, the size p (in bytes) of the telehaptic fragment multiplexed per milli-second is given by

$$p = \frac{D}{8} \cdot 1ms.$$

²If there is no audio/video data, as is the case in the communication from the OP to the TOP (see Fig. 1), then each telehaptic fragment is composed of a single haptic sample.

³Throughout this article, we will use audio/video samples and audio/video frames interchangeably.

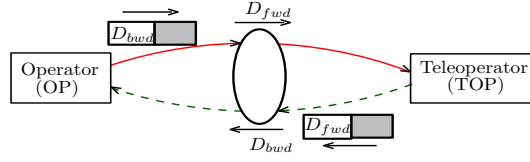


Fig. 2: A schematic representation of the working of in-header delay notification mechanism. D_{fwd} and D_{bwd} indicate the end-to-end delays on the forward and backward channels, respectively.

Due to the mandatory haptic sample in each telehaptic fragment, the size of audio/video data in a fragment is given by

$$s_m = p - s_h.$$

Since audio has a stricter QoS constraint than video, our multiplexer gives audio data priority over video data. That is, in each telehaptic fragment, the multiplexer packs s_m bytes of audio/video data (not previously multiplexed), giving strict priority to audio over video.

2.3. Network Feedback Module

The network feedback module performs two functions: i) it monitors the delays on the forward and backward channels separately through *in-header delay notification* mechanism, and ii) it generates triggers for the respective transmitters to adapt their data rates. We explain these functions in the following.

2.3.1 In-header delay notification We exploit the bidirectional nature of the telehaptic traffic to convey end-to-end delays on each channel to the respective transmitter without transmitting specialized reports (unlike RTP). The in-header delay notification mechanism inserts the end-to-end delay encountered by the latest received packet into the header of the packet to be transmitted, as shown in Figure 2. In particular, the headers of packets transmitted on the forward channel include the end-to-end delay experienced by the last received packet on the backward channel, and vice-versa. This mechanism enables real-time monitoring of the state of congestion on each channel separately, with a negligible overhead of 3 bytes per packet.

The telehaptic nodes are time-synchronized using Network Time Protocol (NTP) [Mills 1991]. The end-to-end delay encountered by a telehaptic packet received is thus calculated as the difference between the time of reception and the timestamp of the earliest haptic sample embedded in the received packet. Note that merging of multiple telehaptic fragments in a packet is explained in detail in Section 2.4.

The in-header delay notification mechanism is more effective than the report-based notification of RTP for three major reasons. Firstly, the higher rate of delay notifications provides finer details of network changes. This enables the telehaptic nodes to swiftly adapt the telehaptic rate to the changing network conditions. Secondly, our scheme does not transmit specialized packets to convey delay feedback, and thus induces a smaller overhead compared to RTP. Thirdly, the in-header delay notification mechanism estimates the delays on the forward and backward channels separately, enabling each transmitter to adapt its rate based on the state of the corresponding channel.

2.3.2 Generation of rate-adaptation triggers Based on the trend observed in the measured delays on each channel, the network feedback module generates two triggers for the corresponding transmitter. The trigger I_C signals that the channel is getting con-

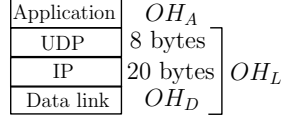


Fig. 3: Representation of UDP/IP stack model, along with the corresponding header size at each layer.

gested; this causes the DPM module to reduce the telehaptic data rate if possible (see Section 2.4). The trigger I_S signals that the channel delays are steady; this causes the DPM module to probe if the channel has spare capacity by increasing the telehaptic data rate if possible (see Section 2.4).

In order to trace the delay pattern, we use an exponentially weighted moving average filter defined by

$$d_{avg}(n) = \alpha * d(n) + (1 - \alpha) * d_{avg}(n - 1), \quad (2)$$

where $0 < \alpha < 1$. Here, $d(n)$ denotes the n 'th end-to-end delay measurement.⁴

The network feedback module generates triggers as follows. The trigger I_C is generated on observing N continuous increasing measurements in $d_{avg}(\cdot)$. Note that a steady increase in the end-to-end delays indicates that queues in the network are building up due to congestion. The trigger I_S is generated if the most recent N entries in $d_{avg}(\cdot)$ satisfy two conditions: (i) the entries exhibit neither an increasing nor a decreasing trend, and (ii) the latter $N - 1$ entries are within a tolerance interval of 10% around the first. Note that generation of the trigger I_S signals that network conditions are steady. It is worth mentioning that since the generation of triggers is based on a *trend* of the end-to-end delays, the proposed rate adaptation scheme remains robust to time synchronization errors of NTP. In our experiments, reported in Section 4, we set $\alpha = 0.2$, as recommended in [Montgomery 2007], and $N = 10$.

2.4. Dynamic Packetization Module

In this section, we describe the dynamic packetization module (DPM), which adapts the telehaptic data rate based on the triggers generated by the network feedback module. We begin by presenting some calculations that illustrate the extent of telehaptic data rate variation possible by varying the packetization rate.

Assuming Ethernet on the data link layer, the overall overhead per packet due to the link layer ($OH_D = 26$ bytes), IP, and UDP headers equals 54 bytes (see Figure 3). Adding to this the application layer header overhead (OH_A) of 13 bytes (see Appendix A), we arrive at a net overhead of 67 bytes/packet. If we transmit each telehaptic fragment as a separate packet (this corresponds to a 1 kHz packetization rate), this amounts to an overall overhead rate of $OHR = 536$ kbps. For a standard TOP payload rate of 560 kbps (haptic - 96 kbps, audio - 64 kbps, video - 400 kbps), the overhead constitutes a substantial proportion (48.9%) of the telehaptic traffic.⁵

Now, suppose that we merge k consecutive telehaptic fragments into a single packet for transmission. We refer to this scheme as the *k-merge* packetization scheme, and we

⁴Note that the OP (TOP) may receive the same delay measurement multiple times; this can happen if the TOP (OP) makes multiple packet transmissions between successive receptions. To avoid the same delay measurement from resulting in multiple updates in (2), we include a one-bit field named *delay indicator* (field D of Table III in Appendix A) to the packet header. This field is set to 1 in case of a repetitive transmission of a previously computed delay, and 0 in case of the transmission of a newly computed delay.

⁵The overhead represents an even higher proportion (72.09%) of the telehaptic traffic from the OP to the TOP, since the payload is composed of only haptic data.

refer to the special case $k = 1$ as the *no-merge* packetization scheme. The telehaptic data rate R_k corresponding to the k -merge packetization scheme (in kbps) is given by

$$R_k = D + \frac{OHR}{k},$$

where D is the telehaptic payload generation rate given by (1) and OHR denotes the overhead rate under the no-merge scheme. Taking $OHR = 536$ kbps, Figure 4 presents the variation of telehaptic overhead rates and packetization delay for different k -merge schemes. Note that these packetization delays correspond to the earliest haptic sample in the packet. Assuming $D = 560$ kbps, we see that on the backward channel the telehaptic transmission rate for the no-merge scheme equals 1096 kbps, whereas the transmission rate for the 4-merge scheme equals 688 kbps. We observe that there is a substantial scope for losslessly varying the telehaptic transmission rate by controlling the packetization parameter k . Of course, the data rate reduction from increasing k comes at the cost of a higher packetization delay at the source.

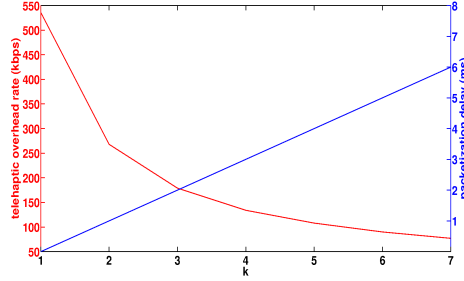


Fig. 4: Telehaptic overhead rate variation for different k -merge packetization schemes, along with the corresponding packetization delay.

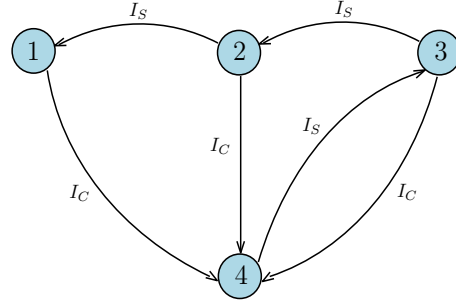


Fig. 5: A finite state transition diagram representation of the step-increase-multistep-decrease approach of DPM with $k_{max} = 4$.

The idea behind DPM is to dynamically adapt the packetization parameter k depending on network conditions. In other words, DPM dynamically switches between different k -merge schemes based on the triggers from the network feedback module. From Figure 4, we note that the overhead reduction becomes insignificant for large values of k , whereas the packetization delay grows linearly in k . Thus, DPM confines the adaptation of k to the range $1 \leq k \leq k_{max}$. In this work, we set $k_{max} = 4$.

DPM is a *step-increase-multistep-decrease* (SIMD) algorithm. This is a variation of the classical additive-increase-multiplicative-decrease (AIMD) congestion control mechanism of TCP [Chiu and Jain 1989]. Specifically, on receiving the trigger I_C (recall that this trigger signals that the network is getting congested), DPM sets $k = k_{max}$. Thus, on sensing congestion in the network, DPM decreases the telehaptic data rate aggressively in order to decongest the network in the shortest possible time. On the other hand, on receiving the trigger I_S (recall that this trigger signals that network delays are steady), DPM decreases k by 1 if $k > 1$. Thus, on sensing that the network is in a steady state, DPM probes if a higher data rate is achievable by decreasing k by one unit. Figure 5 shows a finite state transition diagram representation of DPM.

Note that DPM's dynamic packet rate adaptation will induce additional jitter in the receiver. To get sense of the jitter caused by DPM, we perform the following simple analysis, focusing only on haptic jitter (note that the haptic stream has the tightest

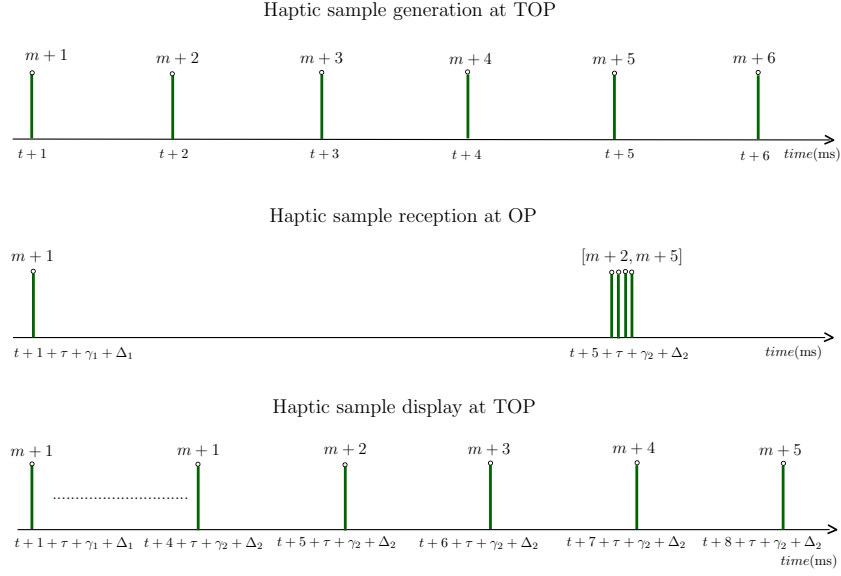


Fig. 6: Timing diagram illustrating haptic sample transmission at TOP, reception and display at OP using zero-order hold strategy. The samples bunched together indicate simultaneous reception due to the 4-merge packet.

jitter constraint). It is easy to see that the maximum jitter occurs when switching from $k = 1$ to $k = k_{max} = 4$. Consider the sequence of haptic samples shown in Figure 6. Suppose that initially, $k = 1$. Note that sample number $m + 1$ is generated at time $t + 1$, and is received and displayed at time $t + 1 + \tau + \gamma_1 + \Delta_1$. Here, τ denotes the one-way propagation delay, and Δ_1 and γ_1 denote the queueing delay and transmission delay of the packet containing sample number $m + 1$, respectively. Now, suppose that starting from sample number $m + 2$, we switch from $k = 1$ to $k = 4$. In this case, sample $m + 2$, which is generated at time $t + 2$, will only get transmitted at time $t + 5$ (along with the next three samples), and will get received and displayed at time $t + 5 + \tau + \gamma_2 + \Delta_2$. Here, Δ_2 and γ_2 denote the queueing and transmission delays, respectively, experienced by the packet containing sample $m + 2$. Thus, the jitter of the haptic sample $m + 2$ equals the difference between its actual display time and its expected display time:

$$(t + 5 + \tau + \gamma_2 + \Delta_2) - (t + 2 + \tau + \gamma_1 + \Delta_1) = 3 + (\Delta_2 - \Delta_1) + (\gamma_2 - \gamma_1).$$

Note that $\gamma_2 \leq 4\gamma_1$. Assuming then that Δ_1 and Δ_2 are comparable, we can bound the jitter by $3(1 + \gamma_1)$. Thus, we see that by restricting k to be at most 4 under DPM, we introduce an additional jitter of at most $3(1 + \gamma_1)$ on the haptic stream. Note that the subsequent 4-merge packet (carrying haptic samples $[m + 6, m + 9]$) arrives at $t + 9 + \tau + \gamma_2 + \Delta_2$. We validate the correctness of the jitter analysis through simulation results reported in Section 4.1.

Overview of Communication Architecture: Figure 7 presents an overview of the proposed telehaptic communication framework. We explain the working with respect to the TOP, whereas similar operations are carried out at the OP as well. On receiving the telehaptic packet at the TOP, the *depacketizer* module decodes the header information. Based on the header contents, the payload is forwarded to the appropriate media display devices. The backward channel delay (D_{bwd}) in the

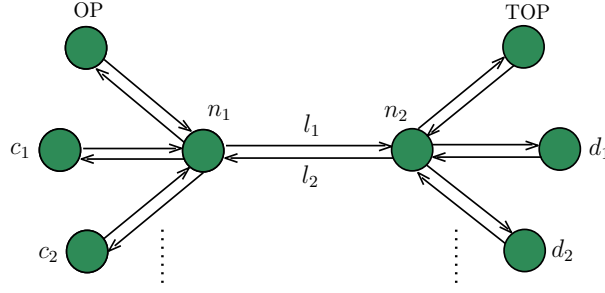


Fig. 8: Single bottleneck dumbbell network topology design for simulation of the telehaptic communication.



Fig. 9: Real-time telepottery experimental setup showing (a) Human operator (b) Teleoperator.

that l_1 and l_2 act as the bottleneck links for the telehaptic traffic on the forward and backward channels, respectively. Thus, queueing delay experienced by the telehaptic application due to network cross-traffic is observable only at the intermediate nodes n_1 and n_2 .

The haptic payload rates on the forward and backward channels are set to 192 kbps and 96 kbps [Gokhale et al. 2015], respectively. The audio and the video payload rates on the backward channel are set to 64 kbps (G.711 PCM standards) and 400 kbps (H.264 standards), respectively. Considering the application layer header sizes (see Appendix A), the no-merge data rate on the forward and backward channels are calculated to be 688 kbps and 1096 kbps, respectively.

Finally, the propagation delay of each link is set to 5 ms. Hence, the one-way propagation delay is 15 ms, which is typically the propagation delay exhibited by a transcontinental link of around 2000 miles. All nodes follow first-in-first-out (FIFO) and drop-tail queueing of packets.

3.2. Perceptual Experiment Setup

It is important to set up a proper experiment to investigate the qualitative effect of DPM on the human multimedia perception, which is not possible through simulations. For this purpose, we conduct a real-time telepottery experiment in which a human subject interacts with a remote, virtual pottery model on a real network through haptic, audio and visual feedback, as described in [Gokhale et al. 2015]. Fig-

ure 9 demonstrates the setup of the telepottery experiment showing the human subject remotely exploring the virtual clay model. The volume preserving pottery model [Chaudhury and Chaudhuri 2014] is rendered at the TOP, and is equipped with a haptic device and a generic webcam. The subject explores and manipulates the virtual clay to design clay pots of his/her choice. The interaction with the remote scene happens through audio-visual feedback and a separate haptic device for the haptic feedback. The master-slave relationship between the two haptic devices is implemented using a proportional-derivative controller; see [Gokhale et al. 2015] for more details.

The subjects were initially briefed about the concept of force feedback as few of them were new to the notion of haptics. Later, we explained them the telepottery task in detail accompanied by a live demonstration of the task. The *training* phase involved the participants performing the task to get acquainted with the telepottery setup. During this phase, the participants explored the telepottery model under an expert's guidance until they were confident of performing the task independently. In order to avoid any perceptual degradation due to the network, the training was performed on a very high bandwidth network, under the no-merge packetization scheme.

After the training, the subjects were moved to a *test* setup consisting of a network emulator tool that allows for configuring the network capacity and propagation delay. Under the emulated network conditions, the subjects independently perform the telepottery task twice: once with no-merge scheme, and once with the proposed DPM scheme. Finally, the subjects were asked to grade the experience of each of the two test experiments, relative to the training, based on three perceptual parameters: *transparency* (the subjects felt as if they were present in the remote virtual environment and are directly interacting with the objects) [Lawrence 1993], *smoothness* (how smooth or jerky is the feedback) [Isomura et al. 2013] and *overall experience*. The grading of each of the three parameters was based on degradation category rating (DCR) [Hoshino et al. 2011; Suzuki and Katsura 2013; Fujimoto et al. 2008] that assigns a subjective scale to a text descriptor in the following manner:

5- imperceptible; 4- slight disturbance, but not annoying; 3- slightly annoying; 2- annoying; 1- very annoying.

For example, the subjects choose 5 if they felt that the degradation in perceptual quality of the test experiments was imperceptible compared to the training phase.

The average training duration was measured to be around 12 minutes, and the average duration for each of the test experiments was around 6 minutes. The subjects had no prior knowledge about the protocol being tested, thereby avoiding grading bias.

3.2.1 System Settings In the real-time telepottery experiment, we use two Phantom Omni haptic devices which capture the low-level dimensions of human haptic perceptual system [Carbon and Jakesch 2013]. Two desktop computers, each with 4 GB RAM and running Windows 7 operating system are employed. The audio-visual information is captured at the TOP using a Microsoft Lifecam VX-2000 webcam. The TOP transmits uncompressed video frames at the rate of 1.2 Mbps (800 kbps more than that in the simulation). However, this change in the video payload rate does not alter the nature of our findings. For these experiments, we increase the channel capacity by 800 kbps compared to the simulation setup to account for the additional video payload.

The real network characteristics are created using a network emulator tool called WANem [Kalitay and Nambiar 2011]. The training phase of the telepottery experiment is performed on a 100 Mbps network. For the testing phase, the emulated channel (both the forward and the backward channels) capacity and one-way propagation delay are configured to 2.3 Mbps and 15 ms, respectively. In the testing phase, we introduce constant bit-rate (CBR) cross-traffic streams of intensity 1.4 Mbps and 400 kbps, respectively, on the forward and the backward channels. In addition, we intro-

duce variable bit-rate (VBR) sources on both channels with intensity $R_{vbr} \in [320, 480]$ kbps with a mean of 400 kbps.

3.2.2 Human Subjects The call for participation in the telepottery task was published on noticeboards in the university. All human subjects who took part in the experiment were either students or faculty members at the university. A total of twenty subjects (ten female and ten male, eighteen right-handed and two left-handed) participated in the perceptual task. The subjects belonged to the age group of twenty three to fifty two years, and none of them suffered from any known neurophysiological disorders. Out of the twenty participants, fourteen were novice haptic users and the rest were regular users of haptic devices. However, all subjects underwent extensive training prior to the test experiments.

4. Experimental Results

In this section, we present a comprehensive experimental evaluation of DPM. Simulation results are presented in Section 4.1, and the results of our perceptual experiments are presented in Section 4.2. In Section 4.3, we compare the performance of DPM with the state of the art in telehaptic communication protocols.

4.1. Simulation Results

In this section, we present the performance evaluation of DPM via simulations. Specifically, we analyze the interplay between DPM and network-oblivious cross-traffic, highlighting DPM's response to highly congested network conditions. For brevity, we present results corresponding to only the backward channel; the performance of DPM on the forward channel is similar.

The simulation begins at time $t = 0$, at which point the telehaptic stream commences transmission. Starting at $t = 0$, we also maintain VBR streams on both channels with intensity $R_{vbr} \in [320, 480]$ kbps with a mean of 400 kbps.⁷ At $t = 500$ ms, we additionally introduce CBR cross-traffic streams on the forward and backward channels. The intensity of the CBR source on the forward channel is fixed at 400 kbps, while the intensity of the CBR source on the backward channel is used as a control parameter to tune the level of congestion on the backward channel. Note that the peak telehaptic data rate on the backward channel equals 1096 kbps (under no-merge packetization), whereas the minimum data rate equals 694 kbps (with $k = 4$ under DPM). Thus, denoting the intensity of the CBR stream on the backward channel by R_{cbr} , we note that when $R_{cbr} > 4$ kbps, the telehaptic stream has insufficient bandwidth to transmit at its peak rate. (Recall that the link capacity equals 1.5 Mbps.) Moreover, $R_{cbr} > 406$ kbps implies that the network is overloaded, since the available capacity is insufficient to even sustain the minimum telehaptic data rate. Thus, the effectiveness of DPM is to be gauged over the range of $R_{cbr} \in [4, 406]$ kbps. In most of our experiments, we set $R_{cbr} = 400$ kbps, which represents a highly congested backward channel. The simulations run for 500 seconds. The throughput, average jitter and packet loss measurements presented in this section are computed after the CBR cross-traffic is switched on, i.e., over the interval $t \in [0.5, 500]$ seconds.

Temporal variation of telehaptic delay: We begin by demonstrating the temporal evolution of the delay experienced by the telehaptic stream under DPM. Figure 10 shows the delay experienced by the haptic samples as a function of the sample generation time, corresponding to $R_{cbr} = 260$ kbps and 400 kbps.

⁷A Skype video-conferencing connection consumes approximately 400 kbps of bandwidth in each direction. Thus, the VBR cross-traffic can be thought of as a video-conferencing stream contending with the telehaptic stream on the bottleneck link.

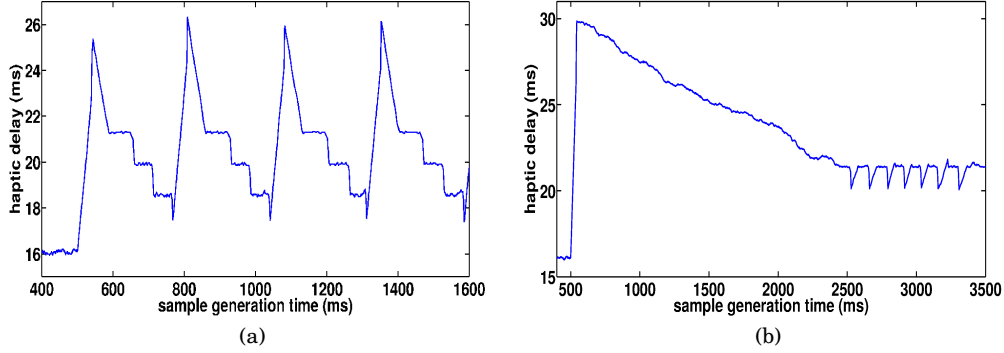


Fig. 10: Haptic delay profile as a result of DPM in presence of CBR cross-traffic (a) 260 kbps (b) 400 kbps.

Let us first consider Figure 10(a). For $R_{cbr} = 260$ kbps, the capacity available to the telehaptic stream on the backward channel equals 840 kbps, which is less than the no-merge transmission rate of 1096 kbps, but more than the 2-merge transmission rate of 828 kbps. Once the CBR source turns on at $t = 500$ ms, the telehaptic stream, initially operating at $k = 1$, sees a rapid delay build-up. DPM responds to this build-up by switching to $k = 4$. This aggressive rate reduction allows the network buffers to drain quickly, avoiding a QoS violation. Once DPM sees a steady delay zone, it probes for a higher telehaptic data rate by decreasing k by 1. But when DPM makes the switch from $k = 2$ to $k = 1$, the instantaneous transmission rate once again exceeds the capacity of the bottleneck link. This in turn leads to a delay build-up, and the cycle repeats.

Figure 10(b) has a similar interpretation. For $R_{cbr} = 400$ kbps, the capacity available to the telehaptic stream on the bottleneck link equals 700 kbps, which is less than the 3-merge transmission rate of 739 kbps, but more than the 4-merge transmission rate of 694 kbps. In this case, the switch from $k = 4$ to $k = 3$ causes a delay build-up, forcing DPM to revert to $k = 4$.

In conclusion, we see that DPM adapts its transmission rate depending on the intensity of cross-traffic it experiences. Moreover, against a steady cross-traffic, DPM results a roughly periodic delay evolution. This is typical of congestion control algorithms; see, for example, [Ha et al. 2008]. Note that even when the backward channel is highly congested (see Figure 10(b)), DPM manages to keep the telehaptic delays below the prescribed QoS limits.

Benefits of step-increase in DPM: Recall that DPM responds to network congestion with an aggressive transmission rate reduction (achieved by a *step-increase* in k to k_{max}), as opposed to a gradual transmission rate reduction (which would be achieved by a *multistep-increase* in k). Figure 11 highlights the benefits of employing the step-increase mechanism over a multistep-increase approach for telehaptic data rate reduction. Specifically, we compare the performance of DPM with an algorithm that increases k by one on receiving the congestion trigger I_C (so long as $k < k_{max} = 4$). For this experiment, we set $R_{cbr} = 400$ kbps. Once the CBR stream starts, the telehaptic stream, initially operating at $k = 1$, experiences a rapid delay build-up due to increased queueing in the network. Note that DPM responds with an aggressive rate reduction ($k = 4$), allowing the network buffers to get flushed quickly, avoiding a QoS violation. On the other hand, the multistep-increase approach cuts the transmission rate in stages, requiring three rate adaptations before setting $k = 4$. As a result, network decongestion occurs much later, leading to a violation of the haptic QoS

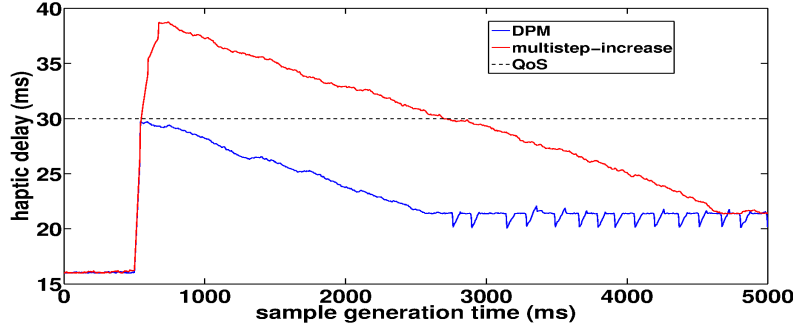


Fig. 11: Haptic delay variation with DPM and multistep-increase approaches, in presence of $R_{cbr} = 400$ kbps.

constraint. Thus, we conclude that DPM's SIMD approach is suitable for congestion control for delay-critical telehaptic applications.

Adaptation to time-varying cross-traffic: In order to test the robustness of DPM to time-varying cross-traffic conditions, we simulate three CBR sources on the backward channel: C_1 , C_2 and C_3 with data rates of 260 kbps, 90 kbps and 50 kbps, respectively. Each of these sources operates over a different interval of time, resulting in an overall cross-traffic scheme shown in Equation (3).

$$R_{cbr} = \begin{cases} 0, & \text{for } 0 < t \leq 500 \text{ ms} \\ 260 \text{ kbps } (C_1), & \text{for } 500 \text{ ms} < t \leq 2500 \text{ ms} \\ 350 \text{ kbps } (C_1, \text{ and } C_2), & \text{for } 2500 \text{ ms} < t \leq 4500 \text{ ms} \\ 400 \text{ kbps } (C_1, C_2 \text{ and } C_3), & \text{for } 4500 \text{ ms} < t \leq 6500 \text{ ms} \\ 0, & \text{for } t > 6500 \text{ ms} \end{cases} \quad (3)$$

Figure 12 shows the temporal variation of DPM source rate. Until 500 ms, DPM achieves its peak rate since the network is uncongested. After 500 ms, the network is unable to support the peak rate, and DPM automatically lowers the telehaptic data rate to avoid congestion. Note that as R_{cbr} increases, DPM lowers its transmission rate progressively. Once the CBR cross-traffic is withdrawn at $t = 6500$ ms, DPM reverts to its peak rate. Thus, we see that DPM performs a robust congestion control under time-varying cross-traffic settings.

| | Max. Delay (ms) | | Max. Jitter (ms) | |
|--------|-----------------|----------|------------------|----------|
| | QoS | Observed | QoS | Observed |
| Haptic | 30 | 29.738 | 10 | 3.628 |
| Audio | 150 | 35.952 | 30 | 5.372 |
| Video | 400 | 73.629 | 150 | 13.255 |

Table I: Comparison of the telehaptic delay and jitter observed for different media for $R_{cbr} = 400$ kbps, along with the corresponding QoS specifications.

Telehaptic delay and jitter measurements: Table I summarizes the observed telehaptic delay and jitter for haptic, audio and video streams, respectively, with $R_{cbr} = 400$ kbps. It can be seen that even under heavy cross-traffic conditions, DPM enables

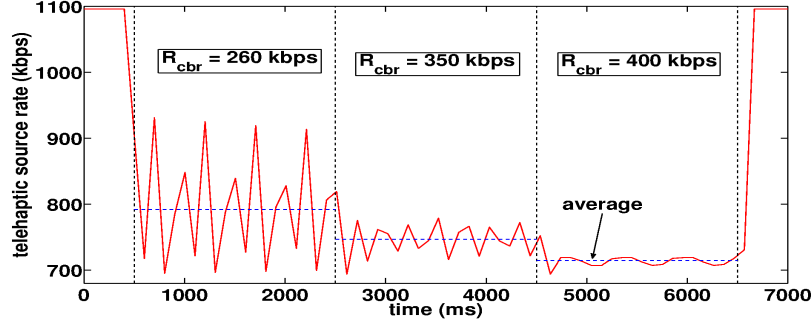


Fig. 12: Telehaptic source rate evolution under time-varying cross-traffic conditions.

the telehaptic application to comply with the QoS limits. Note that the measured haptic jitter is 3.628 ms (see Section 2.4 for an analysis), which is significantly below the QoS jitter limit.

Haptic signal reconstruction: We now study the effects of network cross-traffic, DPM and data extrapolation on the haptic signal reconstruction at the OP. We compare the reconstructed signal with that corresponding to an adaptively sampled strategy, and measure the improvement in haptic signal display that DPM yields. For this purpose, we use real telehaptic traces captured during the telepottery experiment. Ten pilot telehaptic signals were used in the evaluation of the proposed scheme, with each signal corresponding to a different human subject. For brevity, we present results corresponding to a particular pilot signal. We employ a Weber sampler with a threshold of 12% for adaptively sampling the force samples at the teleoperator [Hinterseer and Steinbach 2006]. We use the standard zero-order hold strategy for haptic data extrapolation. For this experiment, we set $R_{cbr} = 400$ kbps.

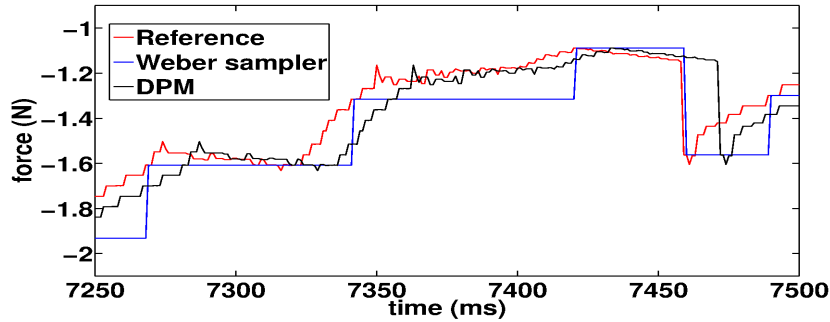


Fig. 13: Graph showing the reconstructed force signals at OP with Weber sampling and DPM for $R_{cbr} = 400$ kbps.

For benchmarking, we make use of a reconstructed signal captured using an ideal (high bandwidth, zero jitter) network; we treat this signal as the reference signal. Figure 13 shows the force signal displayed at the OP under different schemes. As expected, DPM, being a lossless protocol, captures the fine details of the reference

| | SNR (dB) | Improvement over WS (dB) |
|--------------------|----------|--------------------------|
| Weber sampler (WS) | 21.5518 | - |
| DPM | 24.0986 | 2.5468 |

Table II: Comparison of SNR (in dB) in case of Weber sampler and DPM, with $R_{cbr} = 400$ kbps on backward channel.

signal well. On the other hand, the Weber sampled signal is a piecewise constant approximation of the reference signal. It is to be noted that under the Weber sampling strategy, ‘perceptually significant’ samples are displayed earlier at the OP as compared to DPM. This is because of the higher packetization delay under DPM. Using SNR as a performance metric to measure the reconstruction error at the OP (against the reference signal), Table II compares the SNR (in dB) measured for the reconstructed haptic signal under different schemes. We see that DPM exhibits a substantial SNR improvement of around 2.5 dB over the adaptive sampling strategy. In our experiments, we have found a comparable SNR improvement for other haptic traces.

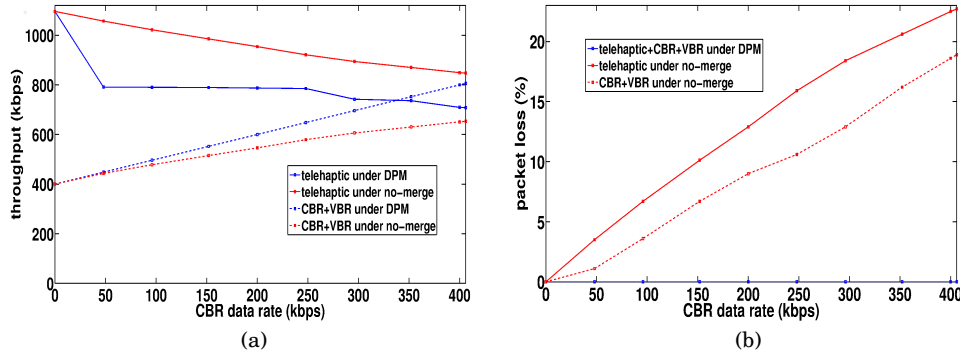


Fig. 14: Telehaptic-CBR traffic interplay demonstrating the improvement of DPM over no-merge in terms of (a) Throughput and (b) Packet loss.

Throughput-Loss Measurements: Figure 14 compares the performance of DPM and the no-merge scheme,⁸ in terms of throughput and packet losses, under various CBR cross-traffic conditions. The results show that for $R_{cbr} < 4$ kbps, the two schemes exhibit similar behavior since the network can sustain the peak telehaptic data rate. As R_{cbr} increases further, the DPM appropriately lowers the telehaptic data rate resulting in zero packet loss until R_{cbr} approaches 406 kbps. On the other hand, the no-merge scheme demonstrates deteriorated performance when $R_{cbr} > 4$ kbps due to its network obliviousness.

Figure 14b shows that the telehaptic and cross-traffic streams sustain severe packet losses with increasing R_{cbr} under the no-merge scheme, whereas DPM avoids packet losses altogether by adapting the telehaptic data rate to the intensity of cross-traffic. We note that DPM is friendly to CBR and VBR cross-traffic. Indeed, the cross-traffic

⁸Recall that the no-merge scheme transmits at the peak telehaptic data rate, oblivious to the state of the network. In the literature, this scheme is also referred to as plain UDP [Postel 1980].

streams see a higher throughput (and zero loss) under DPM as compared to no-merge.

DPM with hold-up: Motivated by Figure 10(a), we propose a variant of DPM that seeks to reduce the jitter induced by frequent rate adaptations. Recall that in the experiment corresponding to Figure 10(a), the maximum data rate for the telehaptic stream that would keep the bottleneck link stable corresponds to $k = 2$. However, when DPM experiences a steady delay at $k = 2$, it switches to $k = 1$, which starts yet another cycle of rate adaptations. In this case, it is clear that if DPM were to hold on to the setting $k = 2$ for a longer period, there would be a reduction in jitter at the receiver. This motivates the following modification of DPM.

DPM with hold-up is identical to DPM, except for the following modification. It remembers the value of k , say \hat{k} , that was operating when the previous I_c trigger was received. Subsequently, once $k = \hat{k} + 1$, the algorithm ignores I_S triggers for a *hold-up duration* T_h .

Note that the hold-up modification would work well under steady or slowly varying cross-traffic conditions. Indeed, if one assumes that the cross-traffic is steady, then one may conclude that the previous I_C trigger was actually caused by the rate adaptation $\hat{k} + 1 \rightarrow \hat{k}$. This suggests that $\hat{k} + 1$ is currently the optimal operating point for the algorithm. Thus, once in this state, *DPM with hold-up* puts off attempts to increase its rate further for a period T_h . Of course, this modification is pessimal in that it misses any opportunities for increasing the transmission rate during the hold-up period T_h .

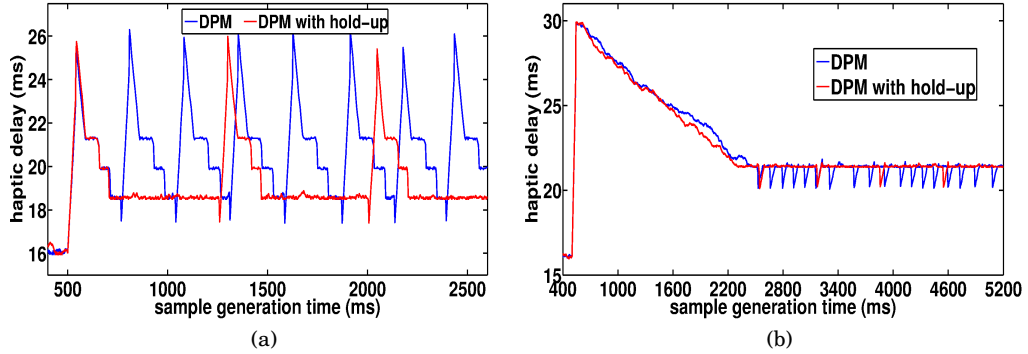


Fig. 15: Haptic delay plots for DPM and DPM with hold-up techniques in presence of CBR cross-traffic (a) 260 kbps (b) 400 kbps.

Figure 15 shows the haptic delay variation plots for DPM with and without hold-up in case of $R_{cbr} = 260$ kbps and 400 kbps. T_h is heuristically chosen to be 500 ms. As expected, under the hold-up modification, the cycles of delay fluctuation occur less frequently. The average jitter for DPM and DPM with hold-up for $R_{cbr} = 400$ kbps are measured as $1.3 \mu s$ and $0.93 \mu s$, respectively. This implies a reduction in average jitter of around 29% over DPM. The SNR of the reconstructed signal under DPM with hold-up is measured to be 24.8332 dB, which is around 0.7 dB higher than the SNR under DPM (see Table II).

In conclusion, when it is known a priori that the cross-traffic is slowly varying, the hold-up modification provides a modest QoS improvement over DPM.

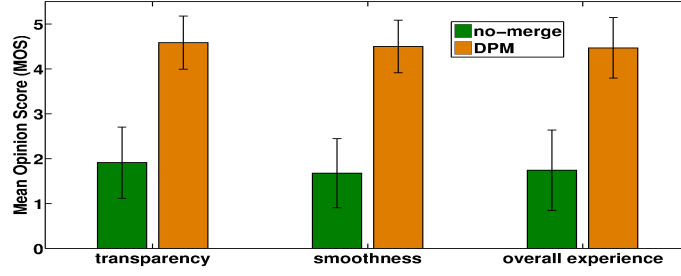


Fig. 16: MOS of the subjective evaluation of the proposed technique on three specific perceptual parameters, averaged over twenty human subjects. The vertical bars denote the standard deviation of the subject grades.

4.2. Telepottery Subjective Grading

We now move to the qualitative results of the real-time telepottery task. Figure 16 presents the mean opinion score (MOS) of the DCR recorded with twenty human subjects for each of the three perceptual parameters i.e., transparency, smoothness and overall experience. We observe that the MOS recorded while using the no-merge technique is less than 2, which corresponds to an annoying user experience. In fact, a few subjects found the no-merge experience so disturbing that they hardly made any contact with the clay model.

When DPM is employed, the MOS under each of the three perceptual parameters improves substantially (in the neighborhood of 4.5, signifying nearly imperceptible degradation in the user experience). This substantiates our claim that the rate adaptation mechanism of DPM introduces negligible perceivable artifacts.

Thus, we conclude that DPM preserves the immersiveness of the telepottery activity in spite of heavy cross-traffic on the network, thereby resulting in a user-friendly and enjoyable telepottery experience.

4.3. Comparison with existing telehaptic communication techniques

In this section, we compare the performance of DPM with RTP [Jacobson et al. 2003] and other recently proposed telehaptic communication protocols.

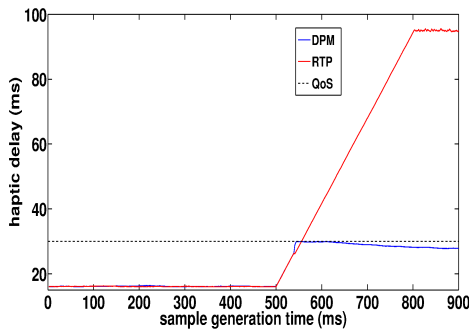


Fig. 17: Early congestion detection and responsiveness of DPM as opposed to sluggish behavior of RTP.

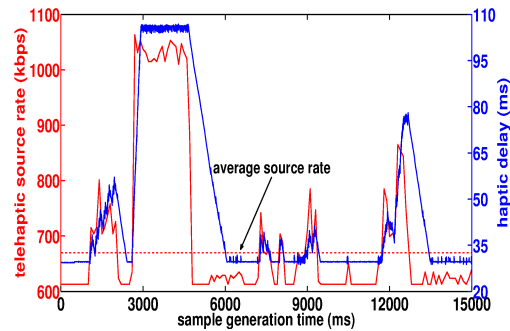


Fig. 18: Telehaptic rate-delay plot of visual-haptic multiplexing on backward channel.

1) Real-time transport protocol (RTP): We begin by comparing DPM with RTP, which is the dominant protocol for media streaming applications on the internet. We use the simulation setup from Section 4.1, with $R_{cbr} = 400$ kbps on the backward channel. Figure 17 shows the variation of the end-to-end delay experienced by haptic samples with the sample generation time. Note that once the CBR cross-traffic is introduced at 500 ms, DPM performs a prompt rate adaptation, maintaining end-to-end delays below the QoS deadline of 30 ms. In the same setting, RTP generates its first and the second RTCP reports at 500 ms and 1000 ms, respectively. Since any rate-control mechanism based on RTP would not make a rate-adaptation prior to 1000 ms, the haptic delays under any such protocol would keep growing as shown in Figure 17, violating the QoS deadlines. Note that network queues build up on the timescale of tens of milliseconds. Thus, for telehaptic applications, RTP, which generates network feedback reports every 500 ms, is too slow to allow for timely rate-adaptation.

2) Visual-haptic multiplexing: We now evaluate DPM against the visual-haptic multiplexing scheme [Cizmekci et al. 2014], which employs the Weber sampler for force updates on the backward channel. For this evaluation, we use the simulation setup from Section 4.1 with $R_{cbr} = 400$ kbps. Figure 18 shows the source rate evolution and the resulting delays under visual-haptic multiplexing obtained using one of the traces from our real-time telepottery experiment. We see that even though the available capacity on the backward channel (700 kbps) exceeds the average transmission rate of the Weber sampler (670 kbps), the instantaneous rate of the Weber sampler fluctuates substantially, resulting in occasional QoS violations; for example, see the interval from 3000 to 6000 ms. Indeed, during times when the operator’s movements are fast, almost every force sample becomes perceptually significant, causing a Weber sampler’s instantaneous transmission rate to far exceed its time-average. It is also worth noting that packet loss measured between 3000 ms and 6000 ms is around 16%, which could potentially lead to significant perceptual degradation. In contrast, under the same network conditions, the results of Section 4.1 show that DPM meets the QoS constraints and results in zero packet loss.

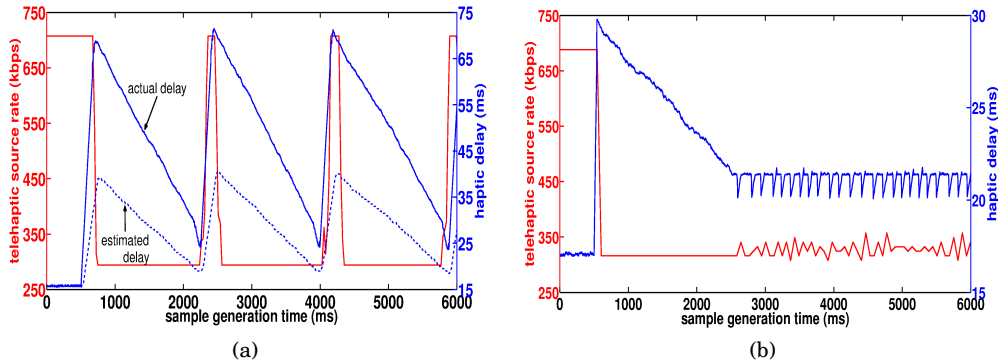


Fig. 19: Telehaptic rate-delay plots for the forward channel with (a) NAFCAH (b) DPM.

3) Network adaptive flow control algorithm for haptic data (NAFCAH): We now compare the performance of NAFCAH [Kokkonis et al. 2015], a protocol that performs RTT-based rate adaptation on the *forward channel*, with DPM. We use the simu-

lation setup from Section 4.1, except that the CBR cross-traffic intensity on the forward channel is increased to 780 kbps; this makes the forward channel highly congested. With the probing packet frequency of NAFCAH set as 100 Hz, Figure 19a shows the evolution of the source transmission rate and the delay experienced by the haptic samples under NAFCAH. Note that NAFCAH incurs substantial QoS violations. The reasons for this are twofold: Firstly, once congestion is detected, NAFCAH cuts its transmission rate in stages (i.e., it employs a *multistep-increase* approach). As discussed in Section 4.1, this results in a relatively sluggish congestion control. Secondly, NAFCAH uses round-trip-time measurements to estimate congestion on the forward channel. This leads to incorrect delay estimations under asymmetric network conditions, as shown in Figure 19a.

In contrast, as seen in Figure 19b, DPM satisfies the QoS constraints well under the same network conditions, thanks to its aggressive *step-increase* mechanism for rate reduction, and its accurate end-to-end delay estimation mechanism.

5. Conclusions and Future Scope

In this paper, we presented DPM, a lossless network-aware application layer protocol for real-time telehaptic communication. In order to enable DPM to quickly respond to network variations, we proposed the network feedback module for communicating the end-to-end delays to the transmitters with negligible overhead. Via extensive simulations, we showed that DPM meets the QoS requirements of telehaptic applications even under highly congested network conditions. We also validated DPM's ability to provide a seamless and immersive user experience over a congested network via a real-time telepottery experiment with human subjects. Finally, we showed that DPM outperforms previously proposed telehaptic communication protocols.

While the present paper explores the interplay between DPM and network-oblivious cross-traffic (CBR/VBR), the interplay between DPM and other network-aware cross-traffic streams represents an interesting avenue for future research. Specifically, we plan to explore the interplay between DPM and a TCP source, and also the interplay between multiple DPM streams. We also intend to carry out a rigorous human subjective grade analysis of DPM in line with [Hamam and El Saddik 2013; Yuan et al. 2014]. Finally, exploring congestion control for multiuser telehaptic settings present an exciting direction for future work.

REFERENCES

- Hussein Al Osman, Mohamad Eid, Rosa Iglesias, and Abdulmotaleb El Saddik. 2007. Alphan: Application layer protocol for haptic networking. In *Proceedings of International Workshop on Haptic, Audio and Visual Environments and Games (HAVE)*.
- Robert Anderson and Mark Spong. 1989. Bilateral control of teleoperators with time delay. *IEEE Trans. Automat. Control* 34, 5 (1989), 494–501.
- Amit Bhardwaj, Onkar Dabeer, and Subhasis Chaudhuri. 2013. Can we improve over weber sampling of haptic signals?. In *Proceedings of Information Theory and Applications Workshop (ITA)*.
- Claus-Christian Carbon and Matrina Jakesch. 2013. A model for haptic aesthetic processing and its implications for design. *Proc. IEEE* 101, 9 (2013).
- Subhajit Chaudhury and Subhasis Chaudhuri. 2014. Volume preserving haptic pottery. In *Proceedings of Haptics Symposium (HAPTICS)*.
- Dah-Ming Chiu and Raj Jain. 1989. Analysis of the increase/decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN Systems* 17, 1 (1989), 1–14.
- Burak Cizmeci, Rahul Chaudhari, Xiao Xu, Nicolas Alt, and Eckehard Steinbach. 2014. A Visual-Haptic Multiplexing Scheme for Teleoperation over Constant-Bitrate Communication Links. In *Haptics: Neuroscience, Devices, Modeling, and Applications*. Springer, 131–138.
- Stella Clarke, Gerhard Schillhuber, Michael Zaeh, and Heinz Ulbrich. 2006. Telepresence across delayed networks: a combined prediction and compression approach. In *Proceedings of International Workshop on Haptic Audio Visual Environments and their Applications (HAVE)*.

- Onkar Dabeer and Subhasis Chaudhuri. 2011. Analysis of an adaptive sampler based on Weber's Law. *IEEE Transactions on Signal Processing* 59, 4 (2011), 1868–1878.
- Mohamad Eid, Jongeun Cha, and Abdulmotaleb El Saddik. 2011. Admux: An adaptive multiplexer for haptic-audio-visual data communication. *IEEE Transactions on Instrumentation and Measurement* 60, 1 (2011), 21–31.
- William Ferrell. 1965. Remote manipulation with transmission delay. *IEEE Transactions on Human Factors in Electronics* 1 (1965), 24–32.
- Elliot Freeman, Alberta Ipser, Austra Palmbaha, Diana Paunoiu, Peter Brown, Christian Lambert, Alex Leff, and Jon Driver. 2013. Sight and sound out of synch: Fragmentation and renormalisation of audio-visual integration and subjective timing. *cortex* 49, 10 (2013), 2875–2887.
- Masaki Fujimoto and Yutaka Ishibashi. 2005. Packetization interval of haptic media in networked virtual environments. In *Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games*.
- Takeshi Fujimoto, Yutaka Ishibashi, and Shinji Sugawara. 2008. Influences of inter-stream synchronization error on collaborative work in haptic and visual environments. In *Proceedings of Symposium on Haptic interfaces for virtual environment and teleoperator systems*.
- Vineet Gokhale, Subhasis Chaudhuri, and Onkar Dabeer. 2015. HoIP: A point-to-point haptic data communication protocol and its evaluation. In *Proceedings of Twenty First National Conference on Communications (NCC)*.
- Vineet Gokhale, Onkar Dabeer, and Subhasis Chaudhuri. 2013. HoIP: Haptics over Internet Protocol. In *Proceedings of International Symposium on Haptic Audio Visual Environments and Games (HAVE)*.
- Vineet Gokhale, Jayakrishnan Nair, and Subhasis Chaudhuri. 2016. Opportunistic adaptive haptic sampling on forward channel in telehaptic communication. In *Haptics Symposium (HAPTICS)*. IEEE.
- Sangtae Ha, Injong Rhee, and Lisong Xu. 2008. CUBIC: a new TCP-friendly high-speed TCP variant. *ACM SIGOPS Operating Systems Review* 42, 5 (2008), 64–74.
- Abdelwahab Hamam and Abdulmotaleb El Saddik. 2013. Toward a mathematical model for quality of experience evaluation of haptic applications. *IEEE Transactions on Instrumentation and Measurement* 62, 12 (2013), 3315–3322.
- Peter Hinterseer, Sandra Hirche, Subhasis Chaudhuri, Eckehard Steinbach, and Martin Buss. 2008. Perception-based data reduction and transmission of haptic data in telepresence and teleaction systems. *IEEE Transactions on Signal Processing* 56, 2 (2008), 588–597.
- Peter Hinterseer and Eckehard Steinbach. 2006. A psychophysically motivated compression approach for 3d haptic data. In *Proceedings of 14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*.
- Peter Hinterseer, E Steinbach, Sandra Hirche, and Martin Buss. 2005. A novel, psychophysically motivated transmission approach for haptic data streams in telepresence and teleaction systems. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.
- Sosuke Hoshino, Yutaka Ishibashi, Norishige Fukushima, and Shinji Sugawara. 2011. QoE assessment in olfactory and haptic media transmission: Influence of inter-stream synchronization error. In *Proceedings of International Workshop Technical Committee on Communications Quality and Reliability (CQR)*.
- Eiichi Isomura, Shuji Tasaka, and Toshiro Nunome. 2013. A multidimensional qoe monitoring system for audiovisual and haptic interactive ip communications. In *Proceedings of Consumer Communications and Networking Conference (CCNC)*.
- Van Jacobson, Ron Frederick, Steve Casner, and H Schulzrinne. 2003. RTP: A transport protocol for real-time applications. (2003).
- Caroline Jay, Mashhuda Glencross, and Roger Hubbard. 2007. Modeling the effects of delayed haptic and visual feedback in a collaborative virtual environment. *ACM Transactions on Computer-Human Interaction (TOCHI)* 14, 2 (2007), 8.
- Hemanta Kumar Kalitay and Manoj K Nambiar. 2011. Designing wanem: A wide area network emulator tool. In *Proceedings of Third International Conference on Communication Systems and Networks (COM-SNETS)*.
- George Kokkonis, Kostas E Psannis, and Manos Roumeliotis. 2015. Network Adaptive Flow Control Algorithm for Haptic Data Over the Internet-NAFCAH. In *Genetic and Evolutionary Computing*. Springer, 93–102.
- Dale A Lawrence. 1993. Stability and transparency in bilateral teleoperation. *IEEE transactions on robotics and automation* 9, 5 (1993), 624–637.
- Seokhee Lee and JongWon Kim. 2007. Haptic event prioritization and network adaptation scheme for collaborative virtual environments. In *Proceedings of Global Telecommunications Conference*.

- Alan Marshall, Kian Meng Yap, and Wai Yu. 2008. Providing QoS for networked peers in distributed haptic virtual environments. *Advances in Multimedia* 2008 (2008).
- Soju Matsumoto, Ichiro Fukuday, Hironao Morinoy, Kenji Hikichiy, Kaoru Sezakiz, and Yasuhiko Yasuda. 2000. The influences of network issues on haptic collaboration in shared virtual environments. (2000).
- David L Mills. 1991. Internet time synchronization: the network time protocol. *IEEE Transactions on Communications* 39, 10 (1991), 1482–1493.
- Dimitrios Miras, Amela Sadagic, Ben Teitelbaum, Jason Leigh, Magda El Zarki, and Haining Liu. 2002. A survey on network QoS needs of advanced internet applications. *Internet2 QoS Working Group* (2002).
- Douglas C Montgomery. 2007. *Introduction to statistical quality control*. John Wiley & Sons.
- ns3. 2011. The network simulator. (2011). <http://www.nsnam.org/>
- Iro Ntonia, Elliot Freeman, Laura Joseph, and Georgia Savva. 2014. Response Efficiency: Behavioural Manifestations of an Emotion-led Subjective Experience of Duration. *Procedia-Social and Behavioral Sciences* 126 (2014), 247–248.
- Jon Postel. 1980. *User datagram protocol*. Technical Report.
- Nizar Sakr, Nicolas D Georganas, and Jiying Zhao. 2011. Human perception-based data reduction for haptic communication in six-dof telepresence systems. *IEEE Transactions on Instrumentation and Measurement* 60, 11 (2011), 3534–3546.
- Eckehard Steinbach, Sandra Hirche, Marc Ernst, Fernanda Brandi, Rahul Chaudhari, Julius Kammerl, and Iason Vittorias. 2012. Haptic communications. *Proc. IEEE* 100, 4 (2012).
- Nobuhiro Suzuki and Seiichiro Katsura. 2013. Evaluation of QoS in haptic communication based on bilateral control. In *Proceedings of International Conference on Mechatronics (ICM)*.
- Uras Tos and Tolga Ayav. 2011. Adaptive RTP Rate Control Method. In *Proceedings of Computer Software and Applications Conference Workshops (COMPSACW)*.
- Raul Wirz, Manuel Ferre, Raul Marín, Jorge Barrio, José M Claver, and Javier Ortego. 2008. Efficient transport protocol for networked haptics applications. In *Haptics: Perception, Devices and Scenarios*. Springer, 3–12.
- Zhenhui Yuan, Shengyang Chen, Gheorghita Ghinea, and Gabriel-Miro Muntean. 2014. User quality of experience of mulsemia applications. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 11, 1s (2014), 15.

APPENDIX

A. Application Layer Frame Structure

In this appendix, we describe in detail the various application layer header fields of the telehaptic packet. Table III shows the proposed application layer frame structure for telehaptic communication. The topmost row is shown for convenience, to indicate the bit positions in the frame. The frame structure starts with the field M . The haptic header size is 8 bytes, whereas the audio and video headers consume 5 bytes each. Since the focus is only on augmenting either audio or video with haptic data, the effective application layer header size is 13 bytes. The audio and video related headers are included only in presence of audio and video payload, respectively. Table IV describes each of the header fields in detail. The telehaptic payload includes haptic-audio/video payload based on the value indicated in the field M . Haptic payload on the forward channel includes position and velocity information of the operator, whereas the backward channel carries force information.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------------------|---|---|---|---|---|---|---|--------------------|---|---|---|---|---|---|---|--------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
| M | | | | | | | | Notification Delay | | | | | | | | | | | | | | | | | | | | | | | |
| Haptic Sample Timestamp | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Audio Frame No. | | | | | | | | | | | | | | | | Audio Payload Size | | | | | | | | | | | | | | | |
| Video Frame No. | | | | | | | | | | | | | | | | Video Payload Size | | | | | | | | | | | | | | | |
| Audio Fragment No. | | | | | | | | Video Fragment No. | | | | | | | | Telehaptic Payload | | | | | | | | | | | | | | | |
| Telehaptic Payload | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table III: Telehaptic packet format at the application layer. The top row is numbered bitwise for better readability.

| Field | Bits | Description |
|--------------------------------------|---------|---|
| M | 3 | Indicates the type of media data contained in the payload. 0: haptic, 1: haptic-audio, 2: haptic-video. The additional bit is included to provide support for additional media types. |
| k | 3 | Indicates the k -merge scheme used for the current packet. |
| D | 1 | Delay indicator field. Indicates the transmission status of the delay embedded in the packet header. 0 - fresh transmission, 1 - repetitive transmission. |
| X | 1 | Reserved for future enhancements to the protocol. |
| Notification Delay | 24 | End-to-end delay inserted by the in-header delay notification mechanism. |
| Haptic Sample Timestamp | 32 | Indicates the generation time (in ms) of the haptic sample in the payload in case of $k = 1$. In case of a higher k , this field indicates generation time of the earliest of the k haptic samples in the payload. |
| Audio Frame No. & Audio Fragment No. | 16 & 8 | Indicates frame number of the current audio payload and fragment number of the current audio frame, respectively. |
| Video Frame No & Video Fragment No. | 16 & 8 | Indicates frame number of the current video payload and fragment number of the current video frame, respectively. |
| Audio & Video Payload Size | 16 & 16 | Indicates the size of the audio and video payload in bytes, respectively. |

Table IV: Detailed description of the application layer header structure.